

Document Title	Assignment 05
Course	CS 200 Introduction to Programming
Academic Year	2016-2017
Semester	Fall
Due Date	December 9, 2016 11:55 pm
Marks	100

The assignment is due on **December 9, 2016 at 11:55 pm**. The late submission policy of 10% deduction per day up to 2 days applies. This assignment will require reasonable amount of time so try to start the assignment as early as you can.

Please keep in mind the following guidelines:

- Do not share your program code with anyone.
- Do not copy code from the internet.
- If you receive any assistance, mention the part of code in which you received assistance.
- You must be able to explain any part of your submitted code.
- All submissions are subject to automated plagiarism detection.

What to submit:

You have to make separate header files, source files and main file. You have to submit all header files (.h files) and all .cpp files containing source code. Zip all .h files and .cpp files into one file named as <your 8 digit roll number>.zip and submit the zip file.

LMS

(100 marks)

You have to implement Learning Management System (LMS) for LUMS.

A user of LMS has three data members: firstName, lastName, password and userName. (may be other data members also). A user can sign in/sign out to LMS system.

A user of LMS can be a

- Student
- Instructor

A student has two additional data members: roll number and status (status can be freshman, sophomore, junior and senior). A student can:

- View his own marks only
- View resources uploaded by instructor
- View roster of a course (list of students enrolled in a course + instructor(s) offering that course)
- View Assignments of a course

An instructor has two additional data members: designation and qualification. An instructor can:

- Add/View Marks of all students enrolled in a course being taught by that teacher
- Add/View Resources to a course tab
- View roster of a course (list of students enrolled in a course + instructor(s) offering that course)
- Add/View Assignments of a course

A course has data members: courseName, courseCode and offeringSchool and capacity. (may be other data members also).

You are free to add data members to any class. Make sure that you can't declare data members of any class **public**.

You have been provided with a file of registered users and five other files: each containing information of offered course, instructor and students enrolled in that course. You have to keep track of students enrolled in a particular course and list of courses in which a particular student is enrolled in.

When a student logs in to LMS, (s)he will see the list of course tabs in which (s)he is enrolled in. On navigating to particular course tab, a student can view list of resources, assignments, view list of students enrolled in that course and can view his/her marks list only.

When an instructor logs in to LMS, he will see the list of courses being taught by him. On navigating to a particular course tab, instructor can view list of resources, add a new resource item to resources list, view list of assignments, add a new assignment to assignment list, add marks of students enrolled in that

course, view marks of all students enrolled in that course and view list of students enrolled in that course. [For adding a new assignment/resource, instructor only adds name of assignment/resource or some text as content. Obviously there will be a due date only for assignments.]

You have to implement concepts of inheritance and polymorphism in this assignment. In your program, you will be having an array of pointers of type user and each pointer will be pointing to an object of child class (of type student or instructor). You can make following virtual functions:

```
viewMarksTab();// a student can only see one option of viewing his own marks only while an instructor can add/view marks of students enrolled in that course
```

```
viewAssignmentsTab(); // a student can only view assignments while an instructor can both add/view assignments
```

```
viewResourcesTab(); // a student can only view resources while an instructor can both add/view resources
```

```
populateUserData();// function to read attributes of user( user can be student or instructor)from file
```

```
// you can also make other virtual functions if you wish to do so
```